

**PATENT APPLICATION**

**CAPTURING SESSION ACTIVITY AS IN-MEMORY SNAPSHOTS  
USING A TIME-BASED SAMPLING TECHNIQUE WITHIN A  
DATABASE FOR PERFORMANCE TUNING AND PROBLEM  
DIAGNOSIS**

Inventor(s): Karl Dias, a citizen of United States, residing at  
910 Crane Avenue  
Foster City, CA 94404

Venkateshwaran Venkataramani, a citizen of India, residing at  
655 Bounty Drive, Apt. 103  
Foster City, CA 94404

Mark Ramacher, a citizen of The United States, residing at  
1619 Gover Lane  
San Carlos, CA 94070

Graham S. Wood, a citizen of United Kingdom, residing at  
P.O. Box 126  
El Granada, CA 94018

Juan Loaiza, a citizen of The United States, residing at  
13 Woodleaf Avenue  
Redwood City, CA 94061

Assignee: Oracle International Corporation  
500 Oracle Parkway, M/S 50P7  
Redwood City, CA 94065

Entity: Large

TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, 8<sup>th</sup> Floor  
San Francisco, California 94111-3834  
Tel: 415-576-0200

Oracle Reference No.: OID-2003-212-01

**CAPTURING SESSION ACTIVITY AS IN-MEMORY SNAPSHOTS  
USING A TIME-BASED SAMPLING TECHNIQUE WITHIN A  
DATABASE FOR PERFORMANCE TUNING AND PROBLEM  
DIAGNOSIS**

5

**CROSS-REFERENCES TO RELATED APPLICATIONS**

[0001] This application is a non-provisional application of and claims benefit to U.S. Provisional Application No. 60/500864, filed September 05, 2003, (Attorney Docket No.: 021756-003701US), which is incorporated by reference in its entirety for all purposes.

10 [0002] The present application incorporates by reference for all purposes the entire contents of the following:

[0003] U.S. Application No. \_\_\_\_\_, entitled "AUTOMATIC DATABASE DIAGNOSTIC MONITOR ARCHITECTURE", Attorney Docket No. 021756-003700US, filed concurrently; and

15 [0004] U.S. Application No. \_\_\_\_\_, entitled "THE TIME MODEL", Attorney Docket No. 021756-004000US, filed concurrently.

**BACKGROUND OF THE INVENTION**

20 [0005] The present invention generally relates to databases and more specifically to apparatus and methods for capturing samples of session activity in a database.

[0006] Enterprise databases continue to grow in size and number resulting in increased systems management and administrative complexity. As the size and complexity of database systems increase, the likelihood that performance problems may result also increases. Analysis of the problems thus becomes an important task to properly maintain database  
25 systems.

[0007] As users connect to a database to request operations, sessions are assigned to the users. A database system may include many sessions that are performing operations in the system at any given time. A session activity log is useful in diagnosing any performance

problems. The log typically includes full traces of all activity performed by a session.

Capturing the log, however, is very expensive and prohibitively slows a database system down.

[0008] Also, because many sessions may be present in a database system over a period of time, it may not be feasible to capture a full trace of activity for all sessions in a database system. In some cases, capturing information may be considered too prohibitive and no information is collected. The activity may be captured using requests externally from the database system, such as SQL requests. This method is again costly and the information returned may not be reliable because the activity that may be causing the performance problem may not be recorded. In other cases, information for only a select number of sessions may be captured. Thus, when analyzing performance problems, a user may not have enough information on session activity to properly diagnose a performance problem.

[0009] Accordingly, methods and apparatus for efficiently capturing information for session activity in a database are needed.

## BRIEF SUMMARY OF THE INVENTION

[0010] Embodiments of the present invention generally relate to capturing snapshots of session activity in a database. In one embodiment, session activity for active sessions is recorded at certain times over a time period. Accordingly, a sample of session activity is taken for sessions in a database.

[0011] In one embodiment, a method for capturing information for activity in a database is provided. The database includes one or more sessions that may or may not be active over a period of time. The method comprises: determining a plurality of times to sample the database; at each of the each plurality of times, performing the steps of: determining one or more active sessions from the one or more sessions included in the database that are active at the time; capturing information for each of the one or more active sessions; and storing the captured information for each of the active sessions.

[0012] In another embodiment, an apparatus for sampling database activity is provided. The database including one or more sessions that may or may not be active over a period of time. The method comprises: a session activity determiner configured to determine one or

more active sessions of one or more sessions in the database at certain times over a time interval, wherein the one or more sessions may or may not be active over a period of time; and an activity sampler configured to capture samples of activity for active sessions at the certain times, wherein the samples are captured by recording a sequence of snapshots of  
5 information for one or more active sessions over time.

[0013] In yet another embodiment, a method for capturing session activity in a database is provided. The database including one or more sessions that may or may not be active over a period of time. The method comprises: determining one or more active sessions from the one or more sessions in the database at certain times over a time interval; and capturing  
10 samples of activity for active sessions at the certain times, wherein the samples are captured by recording a sequence of snapshots of information for one or more active sessions over time.

[0014] Embodiments of the present invention may be included in a computer program product stored on a computer-readable medium.

15 [0015] A further understanding of the nature and advantages of the invention herein may be realized by reference of the remaining portions in the specifications and the attached drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0016] Fig. 1 depicts a system for diagnosing performance problems in a database system  
20 according to one embodiment of the present invention;

[0017] Fig. 2 depicts a system for sampling activity in the database system according to one embodiment of the present invention;

[0018] Fig. 3 depicts a simplified flowchart of a method for capturing information for session histories according to one embodiment of the present invention;

25 [0019] Fig. 4 depicts a more detailed block diagram of a system implementing an embodiment of the present invention;

[0020] Fig. 5 depicts a simplified flow chart of a method for filtering captured information according to one embodiment of the present invention; and

Attorney Docket No.: 021756-004100US

[0021] Fig. 6 is a block diagram of a database system for implementing an embodiment of the invention.

## DETAILED DESCRIPTION OF THE INVENTION

### 5    Overview

[0022] Fig. 1 depicts a system 100 for using a database system according to one embodiment of the present invention. System 100 includes a client 102, a database system 105, and one or more users 108.

10 [0023] In one embodiment, database system 105 includes a database 106 and database server 107. Database server 107 is configured to receive and process requests for database 106. Database system 105 may be any database system and a person skilled in the art will appreciate other components and variations to database system 105. Fig. 6 provides a general description of a database system.

15 [0024] Users 108 send requests for operations to be performed in database 106. The operations include reading data in database 106, writing data to database 106, updating data in database 106, etc. For example, the requests include SQL statements that cause operations to be performed in database 106. The interaction of the users 108 with database 106 using requests is known in the art and a person of skill in the art will appreciate how database systems may be used.

20 [0025] In one embodiment, an automatic database diagnostic monitor (ADDM) may use information captured by embodiments of the present invention. ADDM is described in more detail in the application entitled "AUTOMATIC DATABASE DIAGNOSTIC MONITOR ARCHITECTURE", U.S. Application No. \_\_\_\_\_, Attorney Docket No. 021756-003700US, filed concurrently; and the application entitled "THE TIME MODEL", U.S. Application No.  
25 \_\_\_\_\_, Attorney Docket No. 021756-004000US, filed concurrently, the contents of which are incorporated herein for all purposes. In one embodiment, ADDM uses the information captured to diagnose performance problems in database system 105.

[0026] In addition to ADDM, embodiments of the present invention may be used by other systems. For example, other database systems, application servers, Internet applications, and  
30 any other systems that that require information on session activity may be used.

**Capturing Information for Database Activity**

[0027] Fig. 2 depicts a system 800 for sampling activity in database system 105 according to one embodiment of the present invention. System 800 includes a session activity monitor (SAM) 802, one or more sessions 804, users 108, and a database 806.

5 [0028] SAM 802 is configured to capture information about sessions 804. In one embodiment, SAM 802 captures information without using a query language, such as structured query language (SQL). In one embodiment, SAM 802 is located such that access to information for sessions 804 may be captured internally in database system 105. For example, SAM 802 may be located in database system 105. More specifically, SAM 802  
10 may be located in database server 107 such that database server 107 is effectively capturing information itself. Thus, queries for information are not necessary and may be captured from internal data structures in database system 105.

[0029] Users 108 connect to database system 105. Once connected, a session 804 is generated. Users 108 may then send requests that are processed by database system 105.  
15 The requests and any information associated with operations performed for the requests are associated with the assigned session 804. Accordingly, a session 804 may be any entity that is associated with operations being performed in database system 105. Also, a session 804 may be associated with a user 108 (e.g., a user identifier).

[0030] Sessions 804 may be, over time, active or inactive. For example, in performing an  
20 operation, session 804 may be performing some type of act in database system 105, such as retrieving data. An operation may include many database calls, which are units of work within database system 105. For example, in performing an operation, such as a SQL statement, the operation may include various phases, such as parsing, executing; fetching results, that translate to database calls. A session may be inactive between database calls,  
25 such as data for the phase may not be available yet. A session may also be inactive between operations as a user sees it. For example, the time between when a request ends and a request is received is when a session is inactive as a user sees it. The total time spent in database calls is referred to as “database time or db time”. The time between operation is called “response time” and this is the time the user waits. The key difference between the two is the  
30 time spent on the network. Eventually, the operation may be completed. When the operation

Attorney Docket No.: 021756-004100US

has been completed, additional operations may be performed for session 804 or the session may end.

[0031] Different information may be associated with a session 804. The information may include state information that provides the state of the request at the time. For example, the information includes information describing who a user 108 that requested the operation is (e.g., a user ID), what request (e.g., SQL command) is being processed, what the operation is doing (e.g., waiting, retrieving data, updating data, etc.), and other information. This information is useful when a performance problem is encountered and may be used to diagnose the problem. For example, certain requests and their activity may be viewed, how much time was spent on certain operations can be estimated, etc. and may be used in diagnosing a performance problem. This data may also be used to reconstruct the set of operations/activity in database system 105 for the time under consideration., etc. may be used in diagnosing a performance problem.

[0032] SAM 802 is configured to take snapshots of activity for sessions. In one embodiment, snapshots are combined into a sample. The sample does not include all activity for a session 804. Accordingly, a full trace of information is not taken in one embodiment.

[0033] SAM 802 is configured to capture information from sessions 804 at certain times during a time interval. For example, a time interval, such as every second, may be used for capturing information. At every interval, SAM 802 captures information from sessions 804. Thus, a snapshot of information is captured at certain times rather than a full trace of activity. A sequence of snapshots may be referred to as a sample of the session history. A set of samples do not include all information but may be sufficient to determine the activity for sessions and the database as a whole. For example, if an operation is performing an action for three seconds, a snapshot at every second will capture information for the action. When a new action is started, the next snapshot captures information about the new action. Thus, what a session 804 is doing over a period of time may be determined even though a full trace of information is not recorded. Consumers of the captured information may determine that the information is statistically significant by using well known statistical analysis before drawing conclusions.

Attorney Docket No.: 021756-004100US

[0034] In one embodiment, when SAM 802 determines it is time to capture information, information is captured for active sessions and not inactive sessions. By capturing information for all active sessions at certain time intervals, SAM 802 is able to capture information that may be best for diagnosing performance problems. The burden of capturing information for all sessions 804 that are active is lessened because information is captured at certain intervals. This allows information for all active sessions 804 to be recorded. Also, the burden of capturing information is lessened because information for inactive sessions is not captured. Thus, the captured information may be manageable in that the size captured is smaller and the time taken to capture the information is shorter.

[0035] The information captured may be temporarily stored in memory and then archived in database 806, or the information may be directly stored in database 806. The captured information, whether in temporary memory or on disk, is used in diagnosing problems, etc. Also, the captured information may be selectively filtered where information that is deemed important is stored in database 806.

[0036] Fig. 3 depicts a simplified flowchart of a method for capturing information for session histories according to one embodiment of the present invention. In step 902, SAM 802 determines when it is time to capture activity information in database system 105. In one embodiment, a time interval is determined where information is captured periodically. For example, SAM 802 may capture information every second. Accordingly, a full trace of information is not captured in one embodiment. Rather, a sample of activity is captured where information is captured at certain time intervals.

[0037] In step 904, if it is not time to capture information, the method reiterates to step 902. If it is time, the method proceeds to step 906, where one or more active sessions in database system 105 are determined. In one embodiment, information from active sessions and not inactive sessions is captured. An active session is a session that is actively performing an operation at the time. For example, an active session may be performing a database call or a thread in database server 105 may be executing an operation. An inactive session is a session that is not doing work. For example, an inactive session may be in between database calls or waiting for a resource.



Attorney Docket No.: 021756-004100US

[0038] In one embodiment, a flag is set that indicates that a session 804 is active. SAM 802 then captures information for all active sessions 804. In one embodiment, a flag is set that indicates that a session 804 is active.

[0039] Although it is described that only information from active sessions are captured, it will be understood that information from inactive sessions may be captured. For example, if the number of sessions present in database system 105 is low, then information may be captured from the inactive sessions. In other embodiments, information is captured for just the active sessions. Because information is captured for active sessions, it may not be prohibitive to capture information for all sessions 804 if database system 105 has a lot of sessions 804. Also, typically, when diagnosing a problem, activity for a session 804 is reviewed to determine the problem. Time when a session 804 is inactive may be reviewed also, but the inactive time may be determined from the sequence of captured information. For example, ten snapshots of information may yield information for a session 804 at snapshot one and ten. The session may have been inactive for the time between the first and tenth snapshots and this may be determined by there not being any information for that session 804 in snapshots 2-9.

[0040] In one embodiment, ADDM 104 uses statistical techniques to ensure that the active session samples are statistically significant (from the captured information). For example, monitoring device determines a performance problem. The performance problem indicates which operations may be causing problems. Then, monitoring device looks at the individual requests that were made by users 107 that caused the problem. In order to do this, the samples of active session history are analyzed. The information in different snapshots for an operation are reviewed and a model of what was recorded is developed. The model is useful in determining what a request did in database server 107 (e.g., what operations were performed). If information captured is relatively unintrusive to the operation of database system 105, then snapshots of active sessions may be taken at continuous uniform intervals. The snapshots may provide a statistically significant picture of activity in database system 105. Then, ADDM 104 may use techniques to analyze the activity to determine which operations may be causing performance problems.

[0041] In step 908, information is captured from the one or more active sessions determined in step 906. In one embodiment, internal data structures are scanned to determine

Attorney Docket No.: 021756-004100US

what activity an active session is performing at the time. SAM 802 is then configured to capture the information from the internal data structures without using SQL in one embodiment. The internal data structures may be found in database server 107 and/or database 106. The data structures are read and information is captured.

5 [0042] In step 910, the captured information is stored. In one embodiment, the captured information may first be stored in temporary storage (e.g., memory). The most recent captured information may be stored in temporary storage for easier and faster access. An advantage of this is that the most recent information may be used more often in diagnosing performance problems. The information stored in temporary storage may also be archived in  
10 database 806. This information may be used in future analysis or for analysis over a greater period of time.

[0043] The method then reiterates to step 902 where another snapshot of information is captured. Accordingly, a sequence of snapshots is captured over time. The sequence of information may be very useful in diagnosing a performance problem. Although it is not a  
15 full trace of information, the snapshots may provide a representation of activity in the database that may be just as useful as the full trace. If the activity of a session is known for a periodic interval over a time period, the sequence of information may be pieced together in a statistically significant manner to determine what activity took place. The information may then be used to diagnose a performance problem. Thus, the problem may be diagnosed  
20 without burdening database system 105 with requests for full traces of all activity. Also, information is captured for all active sessions rather than only a select few. A log of information for all active sessions is thus recorded.

[0044] Fig. 4 depicts a more detailed block diagram of a system 1000 implementing an embodiment of the present invention. System 1000 depicts the data flow for data captured by  
25 SAM 802. SAM 802 includes modules, such as session sampler 1006, circular buffer view creator 1008, and database view creator 1010, that may be implemented in software, hardware, or any combination thereof. Users 108 connect to database system 105. Sessions 804 are created and represented by state objects. The state objects include information about the activity for sessions 804.

Attorney Docket No.: 021756-004100US

[0045] Session sampler 1006 captures information from active sessions in sessions 804.

The information is written to a circular buffer 1002. Circular buffer 1002 may temporarily store the captured information. It will be understood that a person of skill in the art will appreciate other storage devices in which information may be written.

5 [0046] Circular buffer 1002 is configured where recent information may be more easily accessible than older information. The recent information may be more likely to be accessed and may be accessed faster than information in database 806. As information for active sessions is captured periodically, it is written to an entry in circular buffer 1002. The information may be written where the oldest information may be flushed from circular buffer  
10 1002. The oldest information in an entry may be deleted or archived to database 806. The filtering process will be discussed in more detail below. In another embodiment, the information in circular buffer 1002 may be flushed periodically (e.g., every 30 seconds) or when there is space pressure (e.g., buffer 1002 is reaching capacity). In this case, entries are not overwritten, rather information is flushed before any entry needs to be overwritten.

15 [0047] A database control 1004 is an interface that may be used to access captured information. In one embodiment, circular buffer view creator 1008 may create a view "V" using information in circular buffer 1002. Also, database view creator may create a view "D" from information from database 806. The information from view "V" and view "D" may be combined to form a complete view. The complete view may then be used to diagnose  
20 performance problems or used for other reasons. The views may be created by time, user 108 or session 804.

[0048] In one embodiment, the captured information is retrieved from circular buffer 1002 and database 806 by references one or more times. For example, captured information may be stored and indexed by time. Information may then be retrieved for a certain time period.  
25 Also, information may be further indexed by user 108 (e.g., which user is associated with a session 804 for which information is stored), by operation performed, etc.

[0049] Fig. 5 depicts a simplified flow chart 1100 of a method for filtering captured information according to one embodiment of the present invention. In step 1102, captured information in circular buffer 1002 is reviewed. In one embodiment, the information is  
30 filtered on a user by user basis. For example, the information captured is associated with a

session 804 and thus a user 108. The information for each session 804 is analyzed to determine if it should be deleted or stored. In another embodiment, the information captured for all active sessions is analyzed as a whole.

[0050] In step 1104, it is determined if the information selected in step 1102 is considered important. In one embodiment, different criteria are used to determine if the information is important and should be saved in database 806.

[0051] In another embodiment, time based filtering may be used. For example, the captured information at certain times is saved, such as one out of every three snapshots of information in circular buffer 1002 is saved.

[0052] In step 1106, if the captured information is not important, then in step 1108, the information is discarded. If the captured information is important, the information is indexed in step 1108.

[0053] In one embodiment, the information is indexed based on the time the information was captured. Thus, if any performance problems are detected, information may be retrieved when the problem occurred. The information may also be indexed by user 108 or session 804. For example, the index may be a user ID for information that was captured about a request a user 108 made. Also, the session 804 that was created for a user 108 may be used as an index. Information specific to users 108 and sessions 804 may then be retrieved. User and session-specific problems may then be diagnosed using this information.

[0054] In step 1110, the indexed information is stored in database 806. The information may be archived for retrieval at a later time. In one embodiment, one difference between information in circular buffer 1002 and database 806 is information in circular buffer 1002 may be accessed faster.

[0055] In one embodiment, information for a session 804 may not be available at a time information is captured. For example, an operation may be in progress for session 804 and still need to finish. A practical example is a surveillance camera taking pictures of a customer in a store. The camera may take a picture of a customer browsing products but that picture does not have the information that may be needed (e.g., whether the product was purchased). In this case, SAM 802 remembers which sessions 804 need information. Once the activity is finished, the previously stored captured information is updated with the new

information. Accordingly, the prior snapshot of information includes the in progress activity and the final activity.

[0056] Accordingly, a sample of information is looking to a future event that may occur.

Information when the event occurs or time that it took to complete is then captured when the

5 event happens. A sample is then completed with the information. This is important because taking snapshots for a sample of database activity means that samples are in progress a lot.

Thus, having a mechanism to determine what information is needed to add to a sample is

important. The information about the completed operation or the time it took to complete is

used in certain analysis of performance problems. In one embodiment, external queries, such

10 as from users 108, cannot perform a data fixup if they are querying database system 105 for information.

[0057] Embodiments of the present invention provide many advantages. For example, the

capturing of information at certain intervals decreases the times information is captured but

increases the amount of information that can be captured. Because information is captured

15 less frequently than for a full trace, information for all active sessions may be captured. The

complete information for all active sessions that is captured at each time interval may be used

to better diagnose performance problems.

[0058] The capturing of information at certain time intervals is more efficient in terms of

CPU efficiency, impact on other database processes, and the total time taken to capture

20 information for session activity. Also, the information is captured without using SQL queries

thereby reducing database activity. Because information is captured at a database level

without querying the database, information that cannot be captured outside the database (i.e.,

by users or applications querying the database) may be captured.

[0059] By having different granularities of snapshots in memory and secondary storage,

25 analysis may be quickly done with more recent data and a more in-depth analysis done with

information in the secondary storage.

[0060] Although embodiments of the present invention were described with reference to a

database system 105, it will be understood that the embodiments are not limited to databases.

For example, embodiments may be used in communications systems that are performing

operations, Internet applications servers, etc. In one embodiment, any system that includes sessions may use embodiments of the present invention.

[0061] Fig. 6 is a block diagram of a system 1200 for implementing an embodiment of the invention. System 1200 includes user computers 1205, 1210, and 1220. User computers  
5 1205, 1210, and 1220 can be general purpose personal computers having web browser applications. Alternatively, user computers 1205, 1210, and 1220 can be any other electronic device, such as a thin-client computer, Internet-enabled mobile telephone, or personal digital assistant, capable of displaying and navigating web pages or other types of electronic documents. Although system 1200 is shown with three user computers, any number of user  
10 computers can be supported.

[0062] A web server 1225 is used to process requests for web pages or other electronic documents from user computers 1205, 1210, and 1220. In an embodiment of the invention, the data analysis software operates within a web browser on a user computer. In this embodiment, all user interaction with the data analysis software is via web pages sent to user  
15 computers via the web server 1225.

[0063] Web application server 1230 operates the data analysis software. In an embodiment, the web application server 1230 is one or more general purpose computers capable of executing programs or scripts in response to the user computers 1205, 1210 and 1215. The web application can be implemented as one or more scripts or programs written in  
20 any programming language, such as Java™, C, or C++, or any scripting language, such as Perl, Python, or TCL.

[0064] In an embodiment, the web application server 1230 dynamically creates web pages for displaying the data analysis software. The web pages created by the web application server 1230 are forwarded to the user computers via web server 1225. Similarly, web server  
25 1225 receives web page requests and input data from the user computers 1205, 1210 and 1220, and forwards the web page requests and input data to web application server 1230.

[0065] The data analysis application on web application server 1230 processes input data and user computer requests and can be stored or retrieved data from database 1235. Database 1235 stores data created and used by the enterprise. In an embodiment, the database 1235 is

a relational database, such as Oracle 9i, that is adapted to store, update, and retrieve data in response to SQL format commands.

[0066] An electronic communication network 1220 enables communication between computers 1205, 1210, and 1215, web server 1225, web application server 1230, and  
5 database 1235. In an embodiment, network 1220 may further include any form of electrical or optical communication devices, including wireless and wired networks. Network 1230 may also incorporate one or more local-area networks, such as an Ethernet network; wide-area networks, such as the Internet; and virtual networks, such as a virtual private network.

[0067] The system 1200 is one example for executing a data analysis software according to  
10 an embodiment of the invention. In another embodiment, web application server 1230, web server 1225, and optionally database 1235 can be combined into a single server computer system. In alternate embodiment, all or a portion of the web application functions may be integrated into an application running on each of the user computers. For example, a Java™ or JavaScript™ application on the user computer is used to retrieve or analyze data and  
15 display portions of the data analysis application.

[0068] While the present invention has been described using a particular combination of hardware and software implemented in the form of control logic, it should be recognized that other combinations of hardware and software are also within the scope of the present invention. The present invention may be implemented only in hardware, or only in software,  
20 or using combinations thereof.

[0069] The above description is illustrative but not restrictive. Many variations of the invention will become apparent to those skilled in the art upon review of the disclosure. The scope of the invention should, therefore, be determined not with reference to the above description, but instead should be determined with reference to the pending claims along with  
25 their full scope or equivalents.